PATENT
83394.0008
Express Mail Label No. EV 325 216 686 US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Art Unit: Not assigned
Hiromichi YAMADA et al.	Examiner: Not assigned
Serial No: Not assigned	
Filed: June 30, 2003	
For: MICRO CONTROLLER FOR PROCESSING COMPRESSED CODES	

TRANSMITTAL OF PRIORITY DOCUMENT

Mail Stop PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Enclosed herewith is a certified copy of Japanese patent application No. 2002-192573 which was filed July 1, 2002, from which priority is claimed under 35 U.S.C. § 119 and Rule 55.

Acknowledgment of the priority document(s) is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

HOGAN & HARTSON L.L.P.

Date: June 30, 2003

By: _____
Anthony J. Orler
Registration No. 41,232
Attorney for Applicant(s)

500 South Grand Avenue, Suite 1900
Los Angeles, California 90071
Telephone: 213-337-6700
Facsimile: 213-337-6701

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００２年　７月　１日

出願番号
Application Number: 特願２００２－１９２５７３

[ST.10/C]: [ＪＰ２００２－１９２５７３]

出願人
Applicant(s): 株式会社日立製作所

２００３年　４月２５日

特許庁長官
Commissioner,
Japan Patent Office 太田信一郎

出証番号　出証特２００３－３０３０６０６

【書類名】　特許願

【整理番号】　HA14471000

【提出日】　平成14年 7月 1日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 9/32

G06F 9/38

【発明者】

【住所又は居所】　茨城県日立市大みか町七丁目１番１号　株式会社日立製作所　日立研究所内

【氏名】　山田　弘道

【発明者】

【住所又は居所】　茨城県日立市大みか町七丁目１番１号　株式会社日立製作所　日立研究所内

【氏名】　安部　雄一

【発明者】

【住所又は居所】　茨城県日立市大みか町七丁目１番１号　株式会社日立製作所　日立研究所内

【氏名】　中塚　康弘

【発明者】

【住所又は居所】　東京都小平市上水本町五丁目２０番１号　株式会社日立製作所　半導体グループ内

【氏名】　山崎　尊永

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社　日立製作所

【代理人】

【識別番号】　100084032

【弁理士】

【氏名又は名称】　三品　岩男

【電話番号】　045(316)3711

【手数料の表示】

【予納台帳番号】　011992

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　圧縮コードを処理するマイクロコントローラ

【特許請求の範囲】

【請求項１】

プログラムにしたがって処理を行なうＣＰＵを備えたマイクロコントローラであって、

プログラムのコードを可変長コードに変換した圧縮コードと、

グループ化されたプログラムのコードについて、各グループの開始アドレスを特定するためのアドレス変換情報と、

各グループごとに、グループに含まれる各圧縮コードのコード長を特定するための圧縮コードタイプ情報とを記憶するメモリと、

ＣＰＵが出力するコードアドレスから、参照すべきアドレス変換情報と圧縮コードタイプ情報とを特定し、特定されたアドレス変換情報と圧縮コードタイプ情報と用いて、対応する圧縮コードアドレスを求め、該当する圧縮コードを前記メモリから読み出す圧縮コード処理部と

を備えることを特徴とするマイクロコントローラ。

【請求項２】

請求項１に記載のマイクロコントローラであって、

前記メモリは、圧縮コードを元のコードに伸張するための辞書情報をさらに記憶し、

前記圧縮コード処理部は、前記辞書情報を参照して、読み出した圧縮コードを元のコードに伸張することを特徴とするマイクロコントローラ。

【請求項３】

請求項１に記載のマイクロコントローラであって、

前記圧縮コード処理部は、前記メモリ中の圧縮コードを記憶した領域と、アドレス変換情報を記憶した領域と、圧縮コードタイプ情報を記憶した領域とを識別するための情報を記憶していることを特徴とするマイクロコントローラ。

【請求項４】

請求項３に記載のマイクロコントローラであって、

前記メモリは、プログラムのコードのブロックの順番に前記アドレス変換情報を記憶し、

プログラムのコードの順番に圧縮コードタイプ情報を記憶することを特徴とするマイクロコントローラ。

【請求項５】

請求項２に記載のマイクロコントローラであって、

前記辞書情報は、対応する圧縮コードのコード長ごとに領域を分けて記憶され、それぞれの領域は、対応する圧縮コードの符号の順番に辞書情報が記憶されることを特徴とするマイクロコントローラ。

【請求項６】

請求項５に記載のマイクロコントローラであって、

前記圧縮コード処理部は、圧縮コードタイプ情報から参照すべき辞書情報が記憶されている領域を特定し、圧縮コードをもとに、特定された領域に含まれる参照すべき辞書情報を特定することを特徴とするマイクロコントローラ。

【請求項７】

請求項１に記載のマイクロコントローラであって、

前記圧縮コード処理部は、圧縮コードを読み出す際に、読み出すべき圧縮コードを含む所定のサイズの圧縮コード群を前記メモリから読み出すものであり、

直前に用いたアドレス変換情報と、圧縮コードタイプ情報と、圧縮コード群とをそれぞれ一時的に記憶する領域を備え、

ＣＰＵが出力したコードアドレスが、直前に読み出した圧縮コードと同一のブロックに含まれる場合には、前記領域に一時的に記憶しているアドレス変換情報と圧縮コードタイプ情報とを用い、

さらに、ＣＰＵが出力したコードアドレスに対応する圧縮コードが、直前に読み出した圧縮コード群に含まれる場合には、前記領域に一時的に記憶している圧縮コード群から圧縮コードを読み出すことを特徴とするマイクロコントローラ。

【請求項８】

請求項１に記載のマイクロコントローラにおいて、

前記圧縮コードには、元のコードと同じコードが含まれることを特徴とするマ

イクロコントローラ。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、マイクロコントローラに係り、特に、メモリに記憶されている圧縮コードを処理するマイクロコントローラに関する。

【０００２】

【従来の技術】

マイクロコントローラ（以下「マイコン」ともいう）は、家電製品、ＡＶ機器、携帯電話、自動車等の機器に組み込まれ、メモリに記憶されているプログラムコードにしたがって処理を行なうことで、それぞれの機器の制御を行なう装置である。

【０００３】

マイコンは、機器に組み込まれて利用されるというその性質上、機器の制御を行なうために必要な性能を備え、かつ、安価であることが要求される。

【０００４】

ところが、近年、マイコンが行なう処理は複雑化しており、処理に必要なプログラムの容量は増加している。このため、マイコンに占めるプログラムメモリの比率が高くなっており、この傾向は今後も続くと考えられている。一般に、プログラムメモリの容量が増えるとその分コストが上昇するため、安価なマイコンを提供するためには、プログラムメモリの容量をいかに抑えるかが重要な問題となっている。

【０００５】

ところで、携帯情報端末等の汎用型の情報処理装置では、プログラムメモリの容量を抑えるために、プログラムコードを圧縮する技術が提案され、実装化も進んでいる。

【０００６】

例えば、ＩＢＭ社が提唱しているＣｏｄｅＰａｃｋと呼ばれるコード圧縮技術は、元のコードを圧縮してプログラムメモリ等に格納する。このとき、圧縮率を

高めるために、圧縮コードは出現頻度等に応じた可変長のビットパターンとしておく。そして、プログラムメモリ等からの読み出し時に、所定のブロック単位で伸長して、伸長後の元のコードをブロック単位でキャッシュメモリに格納するようにしている。

【０００７】

圧縮コードは可変長のため、圧縮後の各コードのメモリアドレスは不規則になる。このため、圧縮コード中に自身のコード長に関する情報を含めるようにしておき、読み出し時には、この情報を元に、各圧縮コードのメモリアドレスを特定して、ブロックの先頭の圧縮コードから順番に伸長するようにしている。

【０００８】

このコード圧縮技術は、ブロック単位で圧縮コードを伸長してキャッシュメモリに格納することから、キャッシュメモリのヒット率が高く、また、プログラムの平均的な実行性能を高くすることが重要とされる汎用の情報処理装置において特に有効と考えられる。

【０００９】

【発明が解決しようとする課題】

しかしながら、従来提唱されているコード圧縮技術は、上述のように、キャッシュメモリの高いヒット率が期待でき、プログラムの平均的な実行性能を高くすることが重要な汎用の情報処理装置には適していても、マイコン等の組み込み機器には、必ずしも最適であるとはいえないと考えられる。

【００１０】

これは以下のような理由による。すなわち、マイコン等の組込み機器では、一般に、プログラムのループ処理が少なく、キャッシュメモリのヒット率は、汎用の情報処理装置ほど高くはならない。このため、キャッシュメモリの書き換えが頻繁に発生することになる。キャッシュメモリの書き換えはブロック単位で行なわれることから、書き換えのたびに、目的のコードを含むブロックの伸長が行なわれる。このとき、目的コード以外のコードは、マイコン等の組込み機器では、必ずしも直後に使用されるとは限らないため、目的のコードを取得するためだけの余分な伸長処理となりかねず、コード伸張時のオーバヘッドが大きくなるから

である。また、組み込み機器では、高いヒット率は期待できないことから、そもそも、キャッシュメモリを備えない場合もあるからである。さらに、マイコン等の組込み機器では、プログラムの平均的な実行性能よりも、制御周期内に決められた処理を確実に実行できるというリアルタイム性が重視されることが多いからである。

【００１１】

したがって、マイコン等の組込み機器では、汎用型の情報処理装置に比べ、目的とする圧縮コードのアドレスをより高速に特定できることが重要となってくる。

【００１２】

本発明の目的は、マイコン等の組み込み機器に好適なコード圧縮技術を提供することにある。

【００１３】

【課題を解決するための手段】

上記課題を解決するため、本発明によれば、

プログラムにしたがって処理を行なうＣＰＵを備えたマイクロコントローラであって、

プログラムのコードを可変長コードに変換した圧縮コードと、

グループ化されたプログラムのコードについて、各グループの開始アドレスを特定するためのアドレス変換情報と、

各グループごとに、グループに含まれる各圧縮コードのコード長を特定するための圧縮コードタイプ情報とを記憶するメモリと、

ＣＰＵが出力するコードアドレスから、参照すべきアドレス変換情報と圧縮コードタイプ情報とを特定し、特定されたアドレス変換情報と圧縮コードタイプ情報と用いて、対応する圧縮コードアドレスを求め、該当する圧縮コードを前記メモリから読み出す圧縮コード処理部と

を備えることを特徴とするマイクロコントローラが提供される。

【００１４】

ここで、前記メモリは、圧縮コードを元のコードに伸張するための辞書情報を

さらに記憶し、

前記圧縮コード処理部は、前記辞書情報を参照して、読み出した圧縮コードを元のコードに伸張することができる。

【作用】

プログラムのコードを可変長コードに変換した圧縮コードと、プログラムのコードをグループ化し、変換後の各グループの開始アドレスを特定するためのアドレス変換情報と、各グループごとに、変換後のグループに含まれる各圧縮コードのコード長を特定するための圧縮コードタイプ情報とをメモリに記憶し、ＣＰＵが出力するコードアドレスから対応する圧縮コードアドレスを直接計算できるように構成することで、マイコン等の組み込み機器に好適なコード圧縮が達成される。

【００１５】

【発明の実施の形態】

本発明の実施の形態について図面を参照して説明する。

【００１６】

図１は、本発明を適用したマイクロコントローラ（マイコン）の実施例の構成の要部を説明するためのブロック図である。

【００１７】

マイコンは、プログラムを実行するＣＰＵ１と、プログラムをあらかじめ圧縮して記憶する領域を含むメモリ２と、ＣＰＵ１からの命令フェッチまたはデータ読み出しの要求に対し、メモリ２から圧縮コードを読み出して元のコードに伸長し、これをＣＰＵ１に供給する圧縮コード伸長回路３とを備えて構成される。

【００１８】

メモリ２は、プログラムのコードを１種以上のコード長の符号に変換した圧縮コードを記憶する圧縮コード領域２３と、ＣＰＵ１が出力するコードアドレスをこれに対応する圧縮コードアドレスに変換するためのアドレス変換情報および圧縮コードタイプ情報をそれぞれ記憶するアドレス変換情報領域２１および圧縮コードタイプ情報領域２２と、圧縮コードを元のコードに変換するための辞書を記憶する辞書領域２４とを含んで構成されている。

【００１９】

ＣＰＵ１は、ＣＰＵアドレスバス１０００にコードアドレスを出力し、ＣＰＵデータバス１００１より伸長されたコードを読み込む。圧縮コード伸長回路３は、メモリバス３０００にアドレスを出力してメモリ２をアクセスし、メモリデータバス３００１より、アドレス変換情報、圧縮コードタイプ情報、圧縮コード、辞書を必要に応じて読み出し、ＣＰＵデータバス１００１に伸長したコードを出力する。

【００２０】

図２は、本発明の基本的な原理を示す第１の実施例における、圧縮コードのフォーマットを説明するための図である。ここでは、４種類のフォーマットの例で説明するが、本発明においては、圧縮コードの種類はこれに限定されるわけではない。また、圧縮する元のコードの単位を１６ビットとして説明を行うが、本発明においてはこれに限定されるわけではない。現状、マイコンで処理される命令やデータのサイズとしては、８ビット、１６ビット、３２ビットが一般的であることから、本実施例における圧縮する元のコードの単位を１６ビットとした。

【００２１】

図２（ａ）は、コード長が４ビットのタイプ０フォーマットを示す。２進表記で００００から１１１１までの１６種類の圧縮コードを定義でき、１６ビットコードを４ビットへと２５％に圧縮することができる。

【００２２】

図２（ｂ）は、コード長が８ビットのタイプ１フォーマットを示す。２進表記で００００００００から１１１１１１１１までの２５６種類の圧縮コードを定義でき、１６ビットコードを８ビットへと５０％に圧縮することができる。

【００２３】

図２（ｃ）は、コード長が１０ビットのタイプ２フォーマットを示す。２進表記で０００００００００から１１１１１１１１１１までの１０２４種類の圧縮コードを定義でき、１６ビットコードを１０ビットへと６２．５％に圧縮することができる。

【００２４】

図２（ｄ）は、コード長が１６ビットのタイプ３フォーマットを示す。このフォーマットは、タイプ０、１、２と異なり、圧縮前のコードをそのまま１６ビットフィールドに持つ。このフォーマットではコードの圧縮は行われない。

【００２５】

タイプ０、１、２のフォーマットの圧縮コードに対しては、これを元のコードに伸長するために辞書を参照することが必要である。

【００２６】

圧縮率を最も高くするという目的のためには、プログラムごとにコードの出現頻度を調べ、出現頻度の高いものから、タイプ０、タイプ１、タイプ２、タイプ３の順に圧縮コードを定義するのが良い。しかしながら、タイプ３フォーマットは辞書を引用する必要がないため、他のタイプに比べて伸長を高速に行える。したがって、出現頻度が高いコードであっても、伸長を高速に行うために、タイプ３フォーマットに割り当てるということも考えられ、本発明においてはコードの出現頻度とタイプの割り当てについて、特に制限を設けるものではない。

【００２７】

図３は、圧縮前のコードと圧縮コードとの対応を説明するための図である。図３（ａ）は、圧縮前のコードのメモリマップの一部を示す図である。図中、ブロックＩ、ブロックＪ、ブロックＫは、圧縮前コードのメモリマップを１６ビット×１６の２５６ビット（３２バイト）単位に分割した領域の一部である。ブロックＪの領域に示すａ～ｐは、それぞれ１６ビットのコードで、ブロックの先頭からの順番を区別するためにアルファベット文字を割り当てたものであり、コードの値とは何ら関係は無い。また、本実施例では、１ブロックに含まれるコードの数を１６として説明するが、本発明においては、この数を１６に限定するものではない。ただし、後述するアドレス変換情報や圧縮コードタイプ情報のアドレス計算を容易にできるなどの理由のために、２のべき乗の数であるほうが望ましい。

【００２８】

図３（ｂ）は、ブロックＩ、ブロックＪ、ブロックＫに対応する圧縮コード領域２３のメモリマップを示す。圧縮コードの各ブロックは、１６ビット境界から

コードが配置されている。ブロックＪの領域については、圧縮コードａ～ｐの配置を示している。圧縮コードは、図２のフォーマットで説明したように、４ビット、８ビット、１０ビット、１６ビットのいずれかのサイズとなる。例えば、ブロックＪの先頭の１６ビットには圧縮コードａ、ｂ、ｃが収まっており、続く１６ビットには圧縮コードｄとｅの一部が収まっている。このように、ブロックの先頭から圧縮コードを１６個連続して配置するため、最後の圧縮コードｐの後には隙間（コードが無い部分）が生じる場合がある。

【００２９】

図３（ａ）圧縮前コードのアドレスと図３（ｂ）圧縮コードのアドレスとの対応は、コード単位ではなく、ブロック単位で管理する。すなわち、圧縮コードの各ブロックの先頭アドレスを保持するようにする。図３では、圧縮前のブロックＩ、ブロックＪ、ブロックＫについて、それぞれの圧縮後のブロックの先頭のアドレスに変換される様子を示している。このブロック単位のアドレス変換情報（圧縮後のブロックの先頭アドレスを示す情報）は、図１に示したようにメモリ２上のアドレス変換情報領域２１に記憶しておく。

【００３０】

図４は、圧縮前コードのブロックとアドレス変換情報との対応を説明するための図である。ここで、圧縮の対象とするコードの容量を最大８メガバイトとして説明をする。図４（ａ）は、圧縮前のコード（最大８メガバイト）のメモリマップを示す。図３で説明したように、圧縮前コードの１ブロックには、１６ビットコードが１６個の３２バイトが含まれるため、ブロックの数は最大で２６２１４４となる。

【００３１】

図４（ｂ）は、アドレス変換情報領域２１のメモリマップを示す。アドレス変換情報は、１ブロックにつき３２ビット（４バイト）の情報を持つものとする。アドレス変換情報領域２１は、メモリ２上のアドレス変換情報ベースアドレスと呼ぶアドレスから、ブロックの順番にアドレス変換情報を連続して記憶するものとする。アドレス変換情報領域２１は最大で、４バイト×２６２１４４の１メガバイトとなる。

【００３２】

図５は、アドレス変換情報のメモリアドレスの計算方法を示す図である。すなわち、ＣＰＵアドレスから、参照すべきアドレス変換情報のアドレスを求めるための計算方法である。ここで、ＣＰＵアドレスは、ＣＰＵ１が出力するコードアドレスである。ＣＰＵ１は、圧縮前のコードアドレスをそのまま出力する。アドレス長は３２ビットで、単位はバイトであるとし、最大で４ギガバイトのアドレス空間を指定することができる。

【００３３】

本実施形態では、メモリ２上の任意の８メガバイト境界からはじまる８メガバイトの領域を、圧縮コード領域２３として設定するするようにする。このため、ＣＰＵアドレスの１０ビット目から続く１８ビット（Ａ２２－Ａ５フィールド）は、コードのブロック番号を示し、最後の５ビット（Ａ４－Ａ０フィールド）は、ブロック内のバイトアドレスを示すことになる。

【００３４】

アドレス変換情報ベースアドレスは、上位１２ビット（Ｂ３１－Ｂ２０フィールド）のみ値を設定可能で、下位２０ビットは全て０固定とし、メモリ２上の１メガバイト境界を指定できるものとする。アドレス変換情報ベースアドレスは、圧縮コード伸長回路３内のレジスタ等に記憶しておく。

【００３５】

アドレス変換情報のメモリアドレスは、アドレス変換情報ベースアドレスのＢ３１－Ｂ２０フィールドと、ＣＰＵアドレスのＡ２２－Ａ５フィールドを使い、本図に示す方法で生成する。すなわち、上位１２ビットをアドレス変換情報ベースアドレスのＢ３１－Ｂ２０フィールドとし、続く１８ビットをＣＰＵアドレスのＡ２２－Ａ５フィールドとし、最後の２ビットを０とする。

【００３６】

次に、図１のメモリ２の圧縮コードタイプ情報領域２２に記憶される圧縮コードタイプ情報について説明する。圧縮コードタイプ情報は、図２で説明した圧縮コードのフォーマットのタイプを識別する情報である。本実施例では、４種類のタイプを識別するために、２ビットのコードを使い、タイプ０は“００”、タイ

プ１は“０１”、タイプ２は“１０”、タイプ３は“１１”とそれぞれ定義するものとする。このように、本実施形態では、圧縮コードのタイプ情報を、圧縮コードとは別に管理するようにしている。

【００３７】

図６は、圧縮コード１ブロックに対する圧縮コードタイプ情報のフォーマットを示す図である。図３と同様に、ブロックの先頭から１６の圧縮コードにａ～ｐと名前を付けると、圧縮コードタイプ情報は、３２ビットの上位から、ａ～ｐというように、圧縮コードと同じ順番に圧縮コードタイプの識別情報を配置する。

【００３８】

図７は、圧縮前コードと圧縮コードタイプ情報との対応を説明するための図である。図７（ａ）は、圧縮前のコード（最大８メガバイト）の仮想メモリマップを示す。圧縮前コードの１ブロックには、１６ビットコードが１６個の３２バイトが含まれるため、ブロックの数は最大で２６２１４４となる。

【００３９】

図７（ｂ）は、圧縮コードタイプ情報領域２２のメモリマップを示す図である。圧縮コードタイプ情報は、１ブロックにつき３２ビット（４バイト）の情報を持つ。圧縮コードタイプ情報領域２２は、メモリ２上の圧縮コードタイプ情報ベースアドレスと呼ぶアドレスから、ブロックの順番に圧縮コードタイプ情報を連続して記憶する。圧縮コードタイプ情報領域２２は最大で、４バイト×２６２１４４の１メガバイトとなる。

【００４０】

図８は、圧縮コードタイプ情報のメモリアドレスの計算方法を示す図である。前述のように、ＣＰＵアドレスのＡ２２－Ａ５フィールドは、コードのブロック番号を示す。

【００４１】

圧縮コードタイプ情報ベースアドレスは、上位１２ビット（Ｂ３１－Ｂ２０フィールド）のみ値を設定可能で、下位２０ビットは全て０固定とし、メモリ２上の１メガバイト境界を指定できるものとする。

【００４２】

圧縮コードタイプ情報のメモリアドレスは、アドレス変換情報ベースアドレスのＢ３１－Ｂ２０フィールドと、ＣＰＵアドレスのＡ２２－Ａ５フィールドを使い、本図に示す方法で生成する。すなわち、上位１２ビットを圧縮コードタイプ情報ベースアドレスのＢ３１－Ｂ２０フィールドとし、続く１８ビットをＣＰＵアドレスのＡ２２－Ａ５フィールドとし、最後の２ビットを０とする。

【００４３】

次に、図１のメモリ２の辞書領域２４に記憶される辞書について説明する。辞書は、図２で説明した圧縮コードを元のコードに変換するためのテーブルである。コード長が４ビットであるタイプ０フォーマットの辞書は１６ビット×１６、コード長が８ビットであるタイプ１フォーマットの辞書は１６ビット×２５６、コード長が１０ビットであるタイプ２フォーマットの辞書は１６ビット×１０２４のテーブルとなる。

【００４４】

図９は、辞書領域２４のメモリマップを示す図である。辞書領域２４は、メモリ２上の辞書ベースアドレスと呼ぶアドレスから、タイプ２、タイプ１、タイプ０の順番に辞書を記憶するものとする。

【００４５】

図１０は、参照すべき辞書のメモリアドレスの計算方法を示す図である。図１０（ａ）に示すように辞書ベースアドレスは、下位１２ビットは全て０固定とし、メモリ上の４０９６バイト境界を指定できるものとする。図１０（ｂ）、図１０（ｃ）、図１０（ｄ）はそれぞれ、タイプ２、タイプ１、タイプ０の辞書アドレスである。辞書領域２４には、それぞれのタイプごとに、圧縮コードの順番（タイプ０を例にすると、０００００、０００１、００１０・・・の順）に元のコードが格納されている。このため、辞書のメモリアドレスは、辞書ベースアドレスのＡ３１－Ａ１２フィールドと、タイプごとのオフセット値と、圧縮コードと、再開ビットの「０」とを結合することによって生成される。ここで、タイプごとのオフセット値は、タイプ２は「０」であり、タイプ１は「１００」であり、タイプ０は、「１０１０００００」となる。

【００４６】

次に、圧縮コード伸長回路３が行なう圧縮コードアドレスの計算処理について説明する。図３に戻って、ＣＰＵ１がブロックＪの先頭から８番目のコードｈを読み出すという例で説明する。まず、ＣＰＵ１が出力したコードアドレスから図５で示したアドレス計算を行い、ブロックＪのアドレス変換情報を読み出す。次に、図８で示したアドレス計算を行い、ブロックＪの圧縮コードタイプ情報を読み出す。圧縮コードタイプ情報により、ブロック内の１６の圧縮コードのそれぞれのサイズを知ることができる。すなわち、タイプ０は４ビット、タイプ１は８ビット、タイプ２は１０ビット、タイプ３は１６ビットである。ブロックＪの先頭から８番目の圧縮コードｈまでの距離（ビット数）は、ブロックＪの圧縮コードの１番目から７番目まで（ａ～ｇ）のサイズを加算したものに等しい。

【００４７】

図１１は、ブロックの先頭から、目的の圧縮コードまでのビット数を計算する回路の例を示す図である。図中ａ～ｏは、ブロックの先頭から１５番目までの圧縮コードのタイプ情報であり、それぞれのタイプ情報を回路３３０において、圧縮コードサイズｓａ～ｓｏに変換する。

【００４８】

図１２は、回路３３０の動作を示す図である。回路３３０は圧縮コードのタイプ情報ｉ（２ビット）をサイズ情報ｏ（４ビット）に変換する。サイズ情報ｏの値は、入力ｉが示すタイプのコード長を２で割った値である。すなわち、タイプ０は２、タイプ１は４、タイプ２は５、タイプ３は８となる。これは、圧縮コードのサイズが２ビットの整数倍であるため、あらかじめ、２で割った（下位方向に１ビットシフトした）データとしておくことにより、図１１の回路全体の規模を小さく抑えることができるからである。

【００４９】

図１１に示した回路３３１は、圧縮コードサイズｓａ～ｓｏと、ＣＰＵアドレスのビット４－１（Ａ［４：１］）を入力し、圧縮コードサイズのうち、目的の圧縮コードに対応する部分とこれ以降のデータとを０にマスクする。ｍａ～ｍｏをマスク後の圧縮コードサイズと呼ぶことにする。

【００５０】

図１３は、回路３３１の動作を示す図である。ＣＰＵアドレスのＡ［４：１］と、マスク後の圧縮コードサイズｍａ～ｍｏの値を示している。ここで、Ｚは４ビット全て０を意味する。ブロックの８番目のコードｈの場合は、Ａ［４：１］＝０１１１となり、ｍａ～ｍｇはそれぞれｓａ～ｓｇ、ｍｈ～ｍｏはＺとなる。

【００５１】

図１１に示した回路３３２は、マスク後の圧縮コードサイズｍａ～ｍｏを加算する回路である。この回路の出力が、ブロックの先頭から目的の圧縮コードまでのブロック内アドレスとなる。

【００５２】

図１４は、回路３３２の例を示す図である。マスク後の圧縮コードサイズｍａ～ｍｏを桁上げ保存加算アレイ（ＣＳＡアレイ）３３２－１で加算し、桁上げ保存形式の出力ｓ、ｒを桁上げ伝播加算器（ＣＰＡ）３３２－２で加算し、ブロック内アドレス（単位は２ビット）を出力する。

【００５３】

図１５は、ＣＳＡアレイ３３２－１の構成例を示す図である。回路３３２－１０は、３入力２出力の桁上げ保存加算器であり、その動作を図１６に示す。ｉ０，ｉ１，ｉ２は、それぞれ１ビットの入力で、ｒは桁上がり（キャリー）出力、ｓは和（サム）出力である。図１５において、マスク後の圧縮コードサイズは、１つのコード（例えばａ）に対して４ビットで表され、ｍａ３，ｍａ２，ｍａ１，ｍａ０のように表されている。図の上段から下段に向かって、各段ごとに１コード分のマスク後の圧縮コードサイズを入力し、出力を次の段に入力する構成となっている。また、図のなかで、桁上げ保存加算回路３３２－１０の入力が０となっているのは、データ０を意味する。ＣＳＡアレイ３３２－１の出力は７ビットのサムｓ６～ｓ０と６ビットのキャリーｒ５～ｒ０となる。図１４のＣＰＡ３３２－２で、ｓ６～ｓ０と、ｒ５～ｒ０を１ビット上位にシフトしたデータの加算を行うことにより、ブロック内アドレスが計算される。

【００５４】

上記の構成において、圧縮コード伸長回路３が、ＣＰＵ１が出力したコードアドレスに対応する圧縮コードのアドレスを特定し、伸長する処理の原理について

説明する。

【００５５】

圧縮コード伸長回路３は、ＣＰＵ１が出力したコードアドレスに対して、図５に示した処理によりアドレス変換情報のアドレスを求めて、メモリ２のそのアドレスを参照することで、目的とする圧縮コードを含むブロックの先頭アドレスを取得する。

【００５６】

そして、図８に示した処理により、圧縮コードタイプ情報のアドレスを求めて、メモリ２のそのアドレスを参照することで、目的とする圧縮コードを含むブロックの圧縮コードタイプ情報を取得する。

【００５７】

取得した圧縮コードタイプ情報から、図１１に示した回路により、目的とする圧縮コードのブロック内におけるアドレスが求められるため、取得したブロックの先頭アドレスを加算することで、目的とする圧縮コードのアドレスを求めることができる。また、圧縮コードタイプ情報から、目的とする圧縮コードの圧縮形式のタイプ（コード長）がわかるため、メモリ２から目的とする圧縮コードを取得することができる。

【００５８】

そして、圧縮形式のタイプにより伸長が必要のない場合には、取得したコードをそのままＣＰＵ１に出力する。一方、伸長が必要な場合には、図１０に示した処理により、辞書のアドレスを求めて、メモリ２のそのアドレスを参照することで、ＣＰＵ１が要求するコードを取得することができる。そして、このコードをＣＰＵ１に出力する。

【００５９】

このように、本実施形態によれば、目的とする圧縮コードのアドレスを直接指定して、その圧縮コードのみを伸長することにより、マイコン等の組み込み機器に好適な、高い圧縮率と、高速な伸長性能とを実現している。

【００６０】

次に、実装上、圧縮コード伸長回路３が目的の圧縮コードを含む圧縮コード群

をメモリ２から読み出す処理について説明する。ここで、圧縮コード群は、メモリ２から読み出すデータの単位（例えば、４バイト分のデータ）である。図１７は、圧縮コードアドレスの計算方法を説明するための図である。図１７（ａ）は、ＣＰＵ１が読み出しを行うコードのアドレス変換情報、すなわち、目的の圧縮コードを含むブロックの先頭アドレスＢＡ［３１：０］（単位はバイト）である。図１７（ｂ）は、目的の圧縮コードの、ブロック内アドレスＩＡ［６：０］（単位は２ビット）である。図１７（ｃ）は、目的の圧縮コードのアドレスＣＡ［３３：０］（単位は２ビット）の計算方法を示している。図に示すように、ＢＡに対し、ＩＡを下位に２ビットシフトして加算を行う。結果アドレスの上位３０ビットＣＡ［３３：４］が、目的の圧縮コードを含む４バイトデータのメモリアドレスを表し、下位４ビットＣＡ［３：０］が、メモリより読み出した４バイトデータにおける目的の圧縮コードの位置を表す。

【００６１】

図１８は、圧縮コード伸長回路３が、メモリ２より読み出した圧縮コード群から、目的の圧縮コードを抽出する処理を説明するための図である。図１７のＣＡ［３３：４］が示すアドレス（単位は４バイト）から読み出した圧縮コード群において、ＣＡ［３：０］が示すビット位置（単位は２ビット）から、圧縮コードのサイズ分を取り出すことにより、目的の圧縮コードｈが得られる。

【００６２】

ここで、メモリ２より読み出した圧縮コード群に、目的の圧縮コードが一部しか含まれていない場合が発生する。例えば、図１８において、圧縮コードｉは、コードの先頭２ビットであり、完全なコードではない。これは、圧縮コードｉのサイズ情報を使うことで検出可能である。この場合には、メモリ２上の次のアドレスにある４バイト分の圧縮コード群を読み出すことにより、目的の圧縮コードを得ることができる。

【００６３】

次に、圧縮コードの伸長に必要なメモリアクセスを効率的に行う方法について説明する。

【００６４】

図１９は、圧縮コードの伸長の処理の流れを説明するためのフロー図である。なお、圧縮コード伸張回路３は、直前に読み出したアドレス変換情報と圧縮コードタイプ情報と圧縮コードアドレスと圧縮コード群とをそれぞれ記憶するレジスタを備えているものとする。

【００６５】

まず、圧縮コード伸長回路３は、ＣＰＵ１が出力するコードアドレスに対し、前回と同じブロックであるかを確認する（Ｓ１０１）。同じブロックであれば、アドレス変換情報と圧縮コードタイプ情報とは、既にメモリ２から読み出し済みであるため次のステップへ移動する。ブロックが異なっていれば、アドレス変換情報と圧縮コードタイプ情報とをメモリ２から読み出す（Ｓ１０２、Ｓ１０３）。

【００６６】

次に、前回読み出した圧縮コード群に目的の圧縮コードが完全に含まれるかどうかを確認する（Ｓ１０４）。含まれていれば次のステップへ移動し、含まれていなければ次の圧縮コード群をメモリ２から読み出す（Ｓ１０５）。このとき、読み出した４バイトの圧縮コード群に目的の圧縮コードが完全に含まれていない場合には、さらに次のアドレスの圧縮コード群も読み出す。

【００６７】

次に、メモリ２から読み出した圧縮コード群から目的の圧縮コード（Ｓ１０６）を抽出する。

【００６８】

そして、圧縮コードが非圧縮形式かどうかを確認する（Ｓ１０７）。非圧縮形式の場合は、コードをそのままＣＰＵ１に出力する。圧縮形式の場合には、辞書アドレスを計算し、メモリ２より辞書を読み出して元のコードに伸長してから（Ｓ１０８）、ＣＰＵ１にコードを出力する。

【００６９】

次に、本発明の第２の実施例について説明する。

【００７０】

図２０は、第１の実施例ではメモリ２に記憶していた辞書を、専用の記憶装置

４に記憶するようにし、圧縮コード伸長回路３は、記憶装置４専用のアドレスバス４０００とデータバス４００１とを使用して、辞書をアクセスする構成を説明するためのブロック図である。例えば、メモリ２が低速、大容量メモリであるとし、メモリ２のアクセスにＣＰＵ１が動作するクロックの数サイクル分を要するとした場合に、小容量で高速な記憶装置４を辞書として使用することにより、圧縮コードの伸長に要する時間を短縮できる。命令フェッチの場合には、分岐や割り込みがない限り、連続したアドレスの命令がフェッチされることになる。アドレス変換情報２１、圧縮コードタイプ情報２２、圧縮コード２３は、メモリから読み出されるデータに、複数コード分の情報が含まれるため、１命令のフェッチに対し、毎回読み出しを行うわけではない。しかし、辞書については、タイプ３以外のフォーマットの圧縮コードの伸長のために、多くの読み出しが必要になる。従って、容量が少なくて済む辞書を、小容量だが高速な記憶装置４に記憶することは、性能の面で有効であり、またハードウエアが多少増加するとしても、これよりはるかに大容量のメモリの削減効果を考えれば、十分実現性があると思われる。

【００７１】

図２１は、本発明の第３の実施例を示すブロック図である。第２の実施例に対して、辞書専用の記憶装置が、辞書１用の記憶装置４と、辞書２用の記憶装置５との２つになり、それぞれが独立したバスで圧縮コード伸長回路３と接続される。この構成は、圧縮の単位に対して、ＣＰＵデータバス１００１の幅が大きい場合に有効になる。例えば、ＣＰＵデータバス１００１が３２ビットで、圧縮の単位が１６ビットの場合を考える。圧縮コード伸長回路３が辞書を１式のみ持つ場合は、圧縮形式であるタイプ０、１、２のフォーマットの圧縮コードは、１サイクルに１つしか伸長できない。タイプ３フォーマットは辞書を引用する必要がないため、他のコードといっしょにＣＰＵデータバス１００１に出力するようにできる。辞書を２式使用すれば、１サイクルに２つの伸長が可能になるため、より伸長の性能が向上することになる。

【００７２】

図２２は、本発明の第４の実施例を示すブロック図である。ここで、ＢＳＣ６

はバスステートコントローラで、ＣＰＵ１の外部へのアクセスを制御する回路である。７はマイコンの階層を表しており、その内部は本発明に関わる主要部のみを示している。外部メモリ８は、マイコン７と外部で接続されるメモリである。この構成では、圧縮コードを外部メモリ８にも記憶する。圧縮コードはメモリ２にも記憶されているが、内蔵メモリ２と外部メモリ８の両方に記憶することも可能であり、外部メモリ８のみに記憶することも可能である。マイコンの内蔵メモリ２だけでは容量が十分でない場合に、外部メモリ８を接続して使用することが考えられるが、外部メモリに圧縮コード８３を記憶することにより、未圧縮のコードを記憶する場合に比べて、性能、消費電力を低減できる可能性がある。それは、マイコン７に読み込まれるコード量が、圧縮形式の方が少なくなるためである。特に、マイコン内部に比べて、外部バスの速度が遅い場合に効果が大きくなる。

【００７３】

次に、圧縮の方法について説明する。図２３は、マイコンプログラムの圧縮の流れを示す図である。圧縮はソースコード９０４、オブジェクトコード９０５の両方に対して行うことができる。プログラム開発環境９００には、コンパイラ９０１、リンカ９０２、圧縮ツール９０３が含まれる。圧縮･非圧縮指定情報９０６とは、ソースコード９０４とオブジェクトコード９０５のどの部分を圧縮し、どの部分を非圧縮にするかの情報である。初期化ルーチン９０７は、マイコンのパワーオンリセット後に、圧縮コードの伸長に必要な情報、例えばアドレス変換情報ベースアドレス、圧縮コードサイズ情報ベースアドレスなどをレジスタ設定するプログラムである。コンパイラ９０１、リンカ９０２、圧縮ツール９０３の処理により、５種類のデータが出力される。圧縮･非圧縮指定情報９０６で、非圧縮に指定した部分は、非圧縮コード９０８として出力される。圧縮に指定した部分は、アドレス変換情報９０９、圧縮コードタイプ情報９１０、圧縮コード９１１、辞書９１２を生成する。これらの出力データをマイコンのメモリに書き込むことにより、非圧縮コード、圧縮コードを含むプログラムの実行が可能になる。

【００７４】

図２４は、非圧縮コードと圧縮コードを混在させる方法を説明するための図である。仮想アドレス空間９２０は、ＣＰＵのメモリマップで、例えば０番地から、内蔵メモリ、外部空間０、外部空間１などいくつかの空間が割り当てられている。これらの空間は非圧縮領域として使う。コンパイル、リンクの処理により、仮想アドレス空間の内蔵メモリ領域９２１の一部に非圧縮対象のコードがマッピングされる。仮想アドレス空間９２０の丁度半分のアドレスから、再び、内蔵メモリ、外部空間０、外部空間１などいくつかの空間が割り当てられており、これらは圧縮領域として使う。コンパイル、リンクの処理により、仮想アドレス空間の内蔵メモリ領域９２２の一部に圧縮対象のコードがマッピングされる。仮想アドレス空間の非圧縮部分９２１は、そのまま実メモリにマッピングされる。圧縮対象部分９２２は、圧縮ツールにより、アドレス変換情報９２４、圧縮コードタイプ情報９２５、圧縮コード９２６、辞書９２７として、実メモリ上にコードが生成される。プログラムの実行時には、ＣＰＵは仮想アドレス空間をアクセスするが、圧縮コード伸長回路は、ＣＰＵアドレスの最上位ビットを見て、これが０であればＣＰＵアドレスをそのまま使って非圧縮コードをアクセスし、１であればアドレス変換を行い、必要に応じて、アドレス変換情報、圧縮コードタイプ情報、圧縮コード、辞書を読み出し、圧縮コードの伸長を行う。

【００７５】

【発明の効果】

上述のように、本発明によれば、マイコン等の組み込み機器に好適なコード圧縮技術が提供される。

【図面の簡単な説明】

【図１】は、本発明を適用したマイクロコントローラの実施例の構成の要部を説明するためのブロック図である。

【図２】は、本発明の基本的な原理を示す第１の実施例における、圧縮コード２３の４つのフォーマットである。（ａ）は、コード長が４ビットのタイプ０フォーマットである。（ｂ）は、コード長が８ビットのタイプ１フォーマットである。（ｃ）は、コード長が１０ビットのタイプ２フォーマットである。（ｄ）は、コード長が１６ビットのタイプ３フォーマットである。

【図３】は、第１の実施例における、圧縮前のコードと圧縮コードとの対応を説明するための図である。

【図４】は、第１の実施例における、圧縮前のコードのブロックとアドレス変換情報のアドレスの対応を説明するための図である。

【図５】は、第１の実施例における、アドレス変換情報のメモリアドレスの計算方法を示す図である。

【図６】は、第１の実施例における、圧縮コード１ブロックに対する圧縮コードタイプ情報を示す図である。

【図７】は、第１の実施例における、圧縮前のコードと圧縮コードタイプ情報のアドレスの対応を説明するための図である。

【図８】は、第１の実施例における、圧縮コードタイプ情報のメモリアドレスの計算方法を示す図である。

【図９】は、第１の実施例における、辞書のメモリマップを示す図である。

【図１０】は、第１の実施例における、辞書のメモリアドレスの計算方法を示す図である。

【図１１】は、第１の実施例における、ブロックの先頭から目的の圧縮コードまでのビット数を計算する回路の例を示す図である。

【図１２】は、第１の実施例における、ブロックの先頭から目的の圧縮コードまでのビット数を計算する回路において、圧縮コードタイプ情報を圧縮コードサイズに変換する回路３３０の動作を示す図である。

【図１３】は、第１の実施例における、ブロックの先頭から目的の圧縮コードまでのビット数を計算する回路において、１ブロックの圧縮コードサイズを目的の圧縮コードに対応する部分とこれ以降を０にマスクする回路３３１の動作を示す図である。

【図１４】は、第１の実施例における、ブロックの先頭から目的の圧縮コードまでのビット数を計算する回路において、マスク後の圧縮コードサイズを加算し、圧縮コードのブロック内アドレスを計算する回路３３２のブロック図である。

【図１５】は、第１の実施例における、回路３３２の構成要素の１つである、桁上げ保存加算アレイ３３２－１の例である。

【図１６】は、第１の実施例における、桁上げ保存加算アレイ３３２－１の構成要素である、桁上げ保存加算器の動作を示す図である。

【図１７】は、第１の実施例における、圧縮コードのアドレス計算方法を示す図である。

【図１８】は、第１の実施例における、メモリより読み出した圧縮コードから、目的の圧縮コードを抽出する方法を示す図である。

【図１９】は、第１の実施例における、圧縮コードの伸長フローを示す図である。

【図２０】は、本発明を適用したマイクロコントローラの第２の実施例の構成の要部を説明するためのブロック図である。

【図２１】は、本発明を適用したマイクロコントローラの第３の実施例の構成の要部を説明するためのブロック図である。

【図２２】は、本発明を適用したマイクロコントローラの第４の実施例の構成の要部を説明するためのブロック図である。

【図２３】は、本発明におけるマイコンプログラムの圧縮の流れを示す図である。

【図２４】は、本発明における非圧縮コードと圧縮コードを混在させる方法を説明するための図である。

【符号の説明】

１…ＣＰＵ

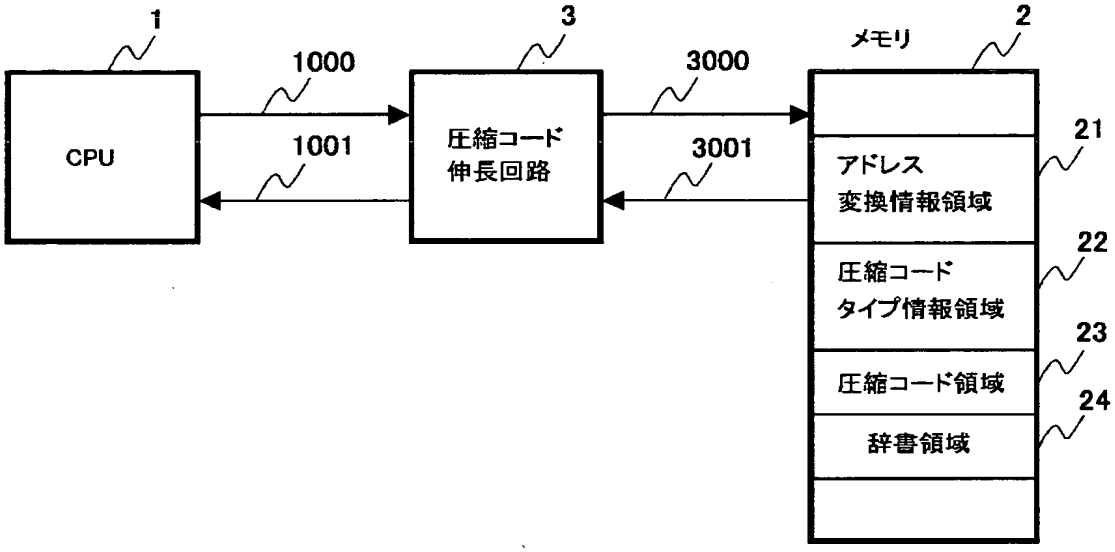
２…メモリ

３…圧縮コード伸長回路

【書類名】　図面

【図１】

図1

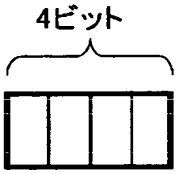


出証特２００３－３０３０６０６

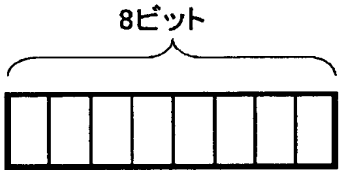
【図２】

図2

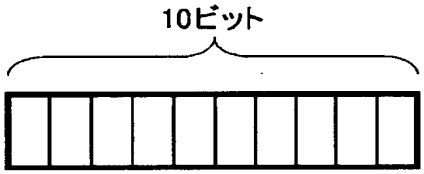
(a)タイプ0
・圧縮形式
・16種



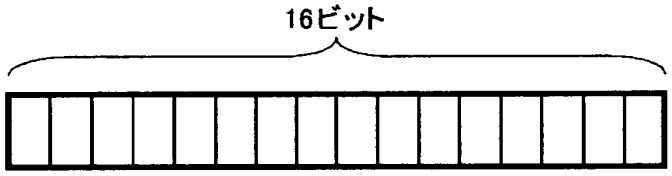
(b)タイプ1
・圧縮形式
・256種



(c)タイプ2
・圧縮形式
・1024種

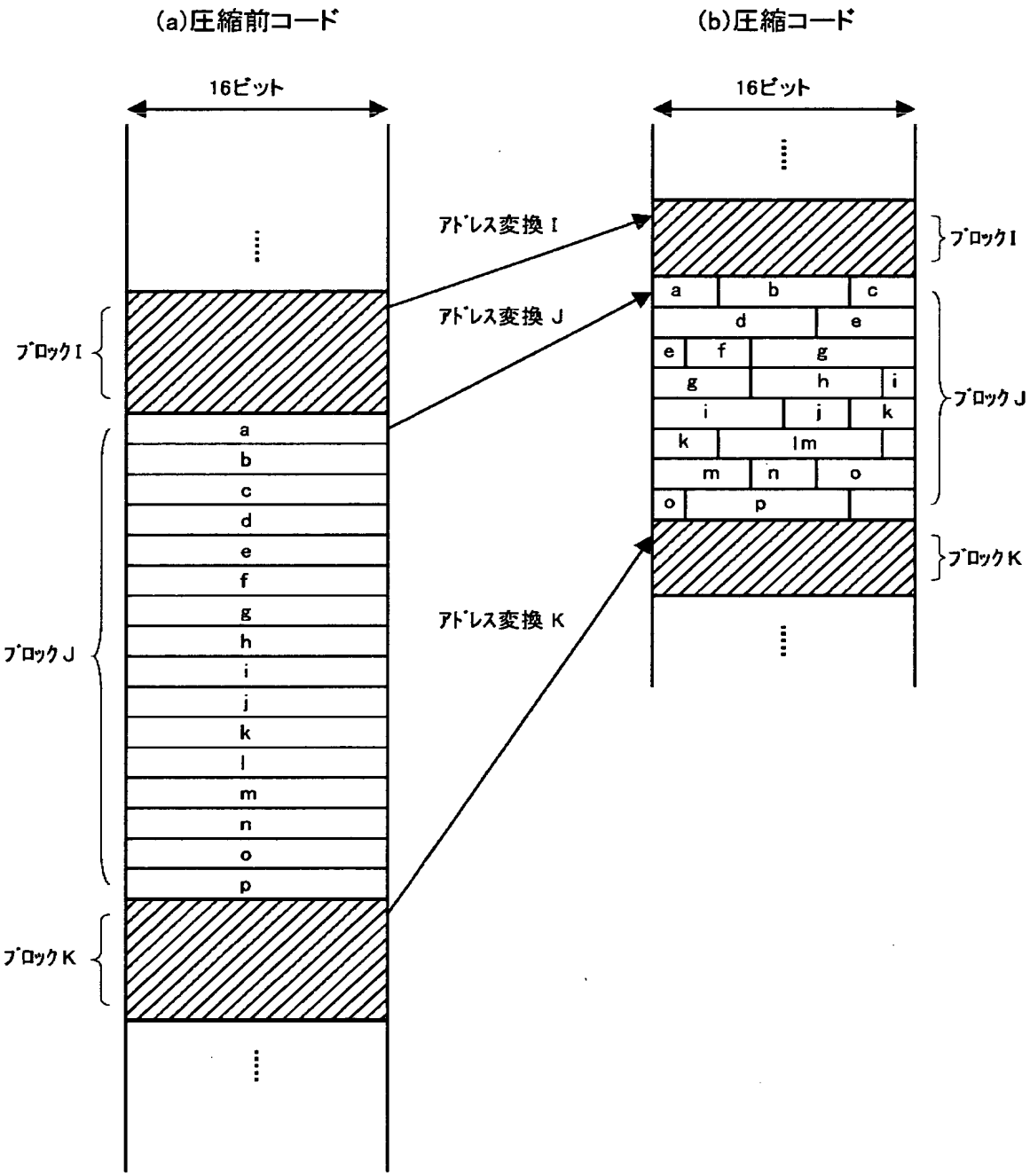


(d)タイプ3
・非圧縮形式
・65536種



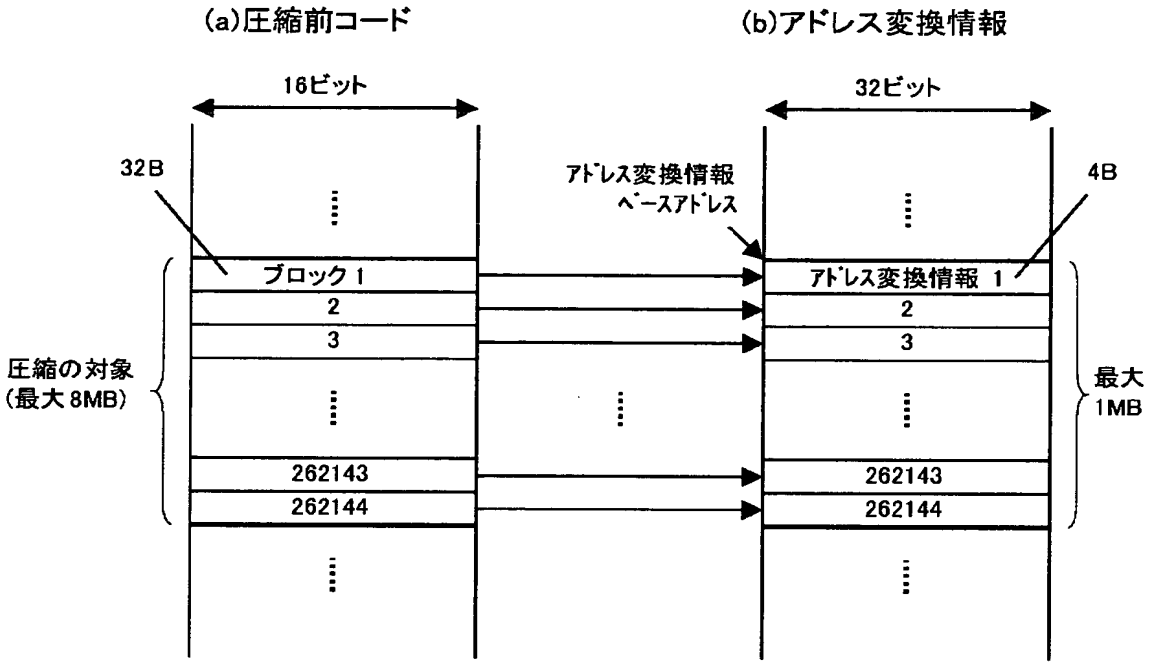
【図３】

図3


(a)圧縮前コード
(b)圧縮コード
16ビット
16ビット
アドレス変換 I
アドレス変換 J
アドレス変換 K
ブロックI
ブロックJ
ブロックK
ブロックI
ブロックJ
ブロックK
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p

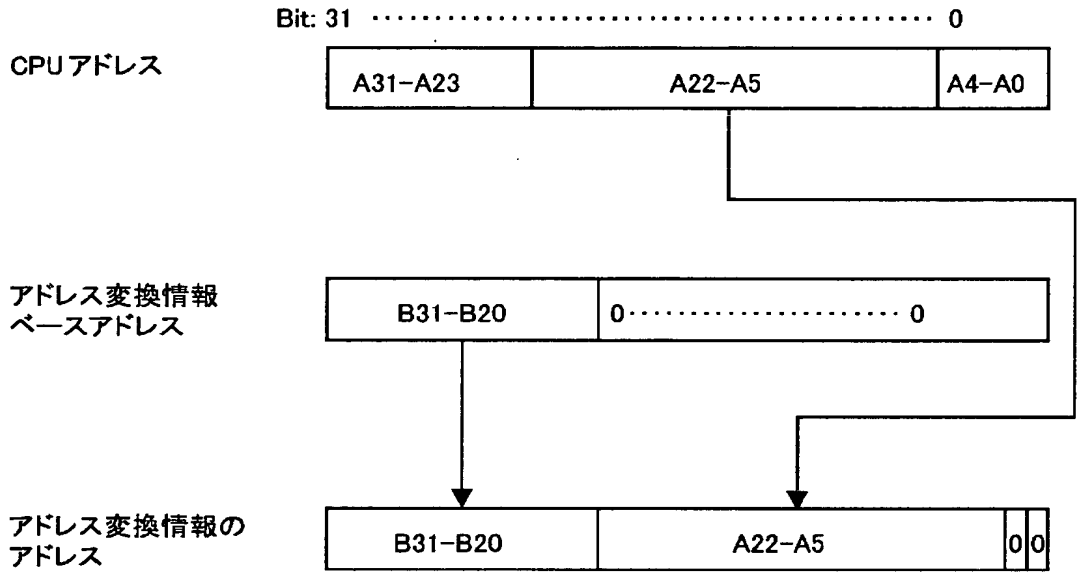
【図４】

図4


(a)圧縮前コード
(b)アドレス変換情報
16ビット
32ビット
32B
4B
アドレス変換情報
ベースアドレス
ブロック１
2
3
262143
262144
アドレス変換情報 1
圧縮の対象
(最大8MB)
最大
1MB

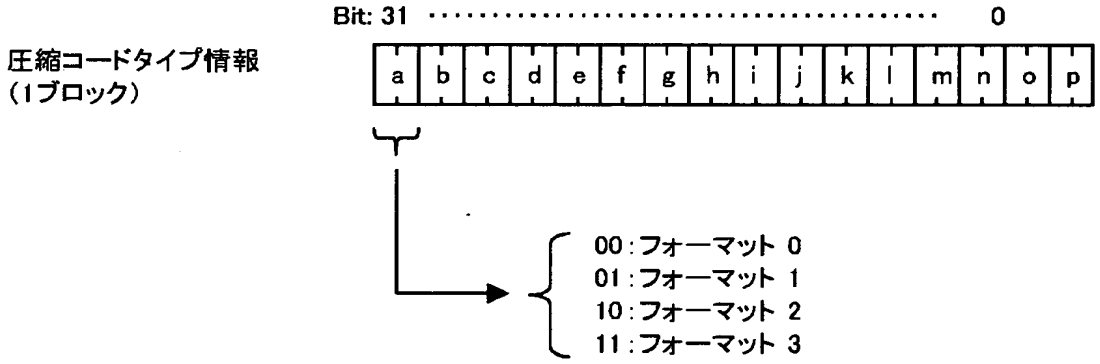
【図５】

図5


Bit: 31
0
CPUアドレス
A31-A23
A22-A5
A4-A0
アドレス変換情報
ベースアドレス
B31-B20
0
0
アドレス変換情報の
アドレス
B31-B20
A22-A5
0
0

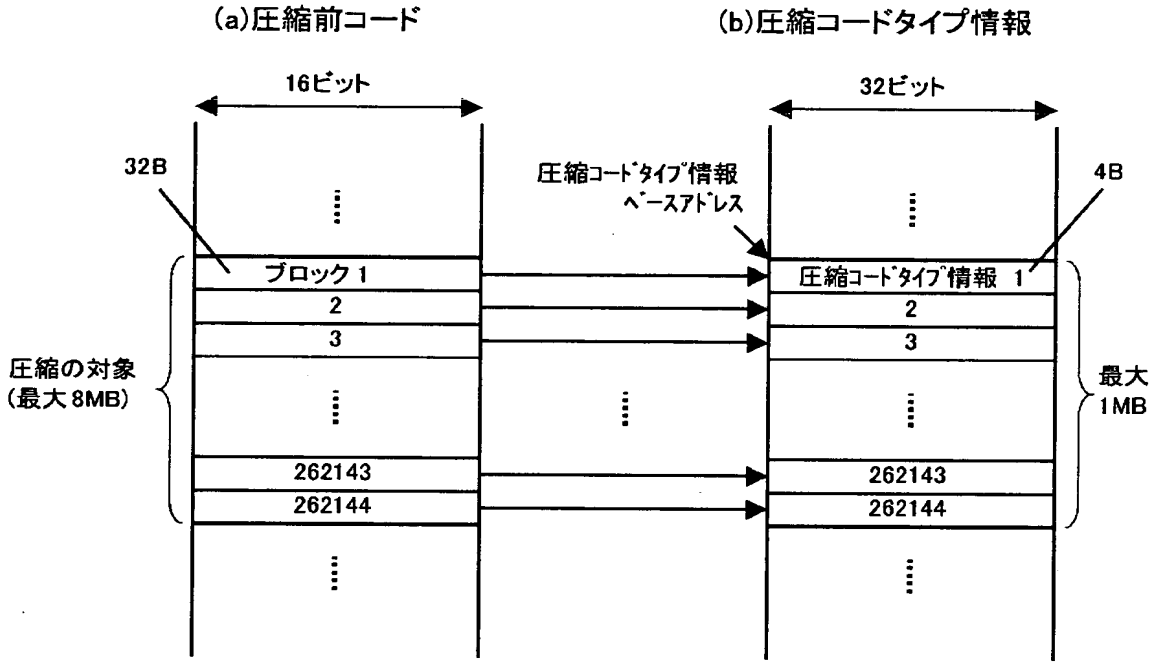
【図６】

図6


Bit: 31
0
圧縮コードタイプ情報
(1ブロック)
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
00:フォーマット 0
01:フォーマット 1
10:フォーマット 2
11:フォーマット 3

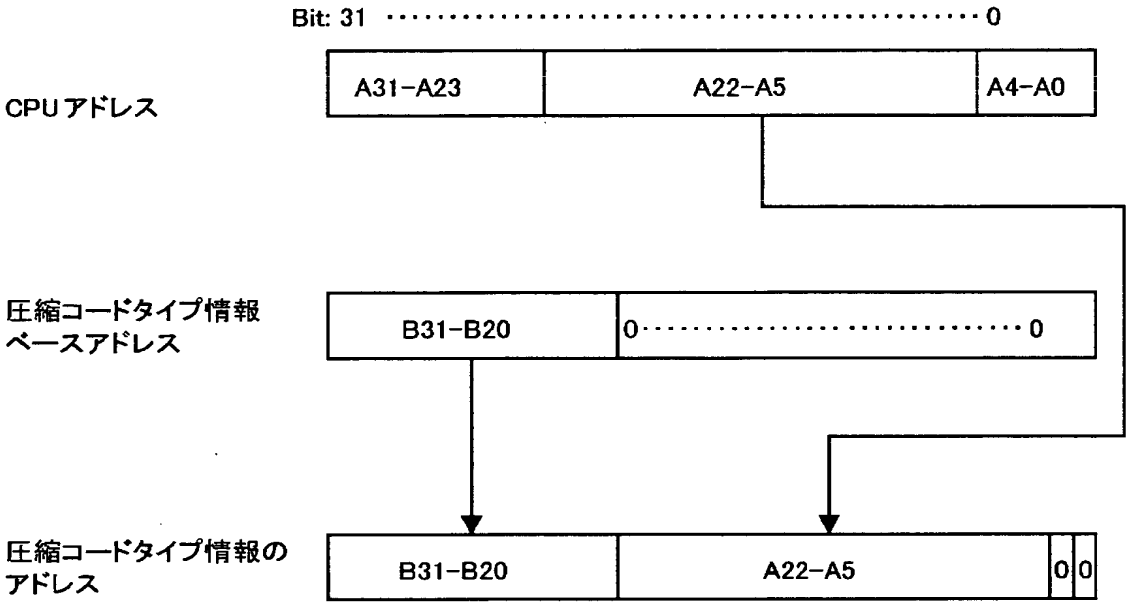
【図７】

図7


(a)圧縮前コード
(b)圧縮コードタイプ情報
16ビット
32ビット
32B
4B
圧縮ｺｰﾄﾞﾀｲﾌﾟ情報
ﾍﾞｰｽｱﾄﾞﾚｽ
ブロック１
2
3
262143
262144
圧縮ｺｰﾄﾞﾀｲﾌﾟ情報 1
2
3
262143
262144
圧縮の対象
(最大8MB)
最大
1MB

【図８】

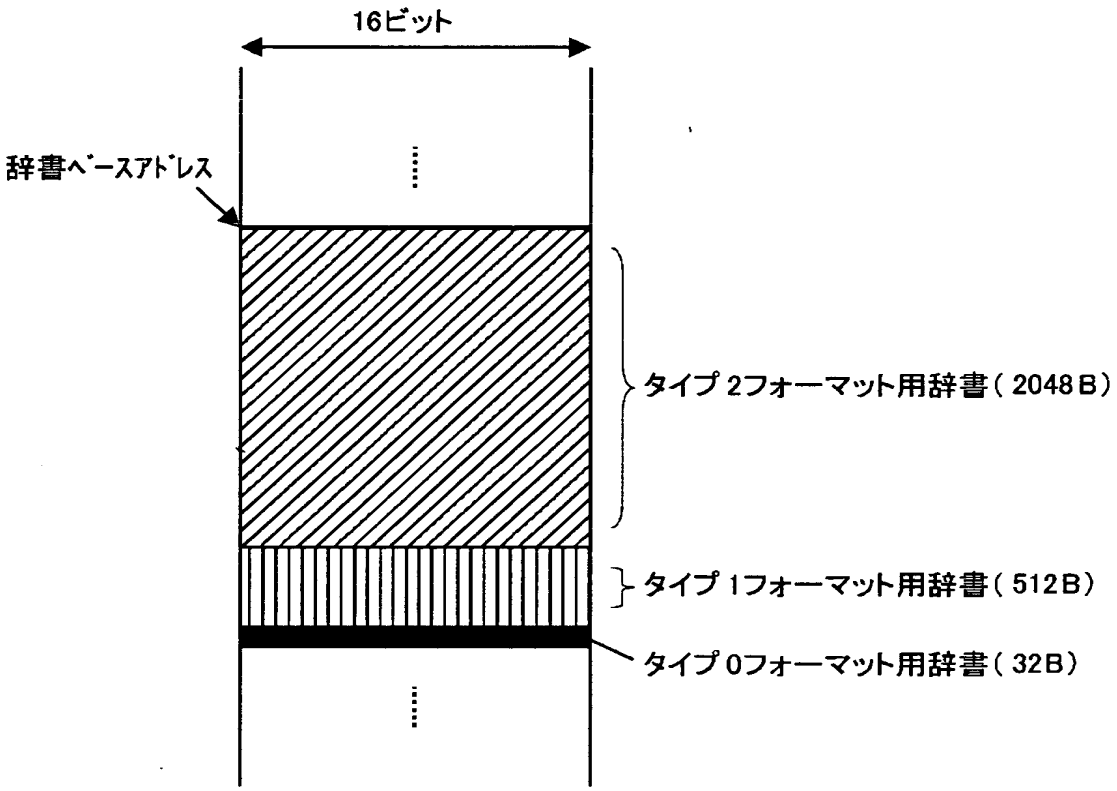
図8



出証特２００３－３０３０６０６

【図９】

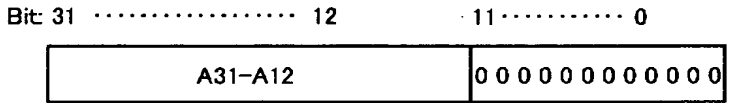
図9


16ビット
辞書ベースアドレス
タイプ2フォーマット用辞書（2048B）
タイプ1フォーマット用辞書（512B）
タイプ0フォーマット用辞書（32B）

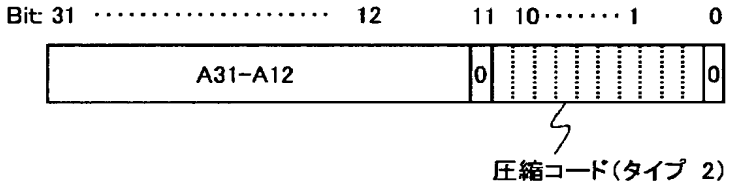
【図１０】

図10

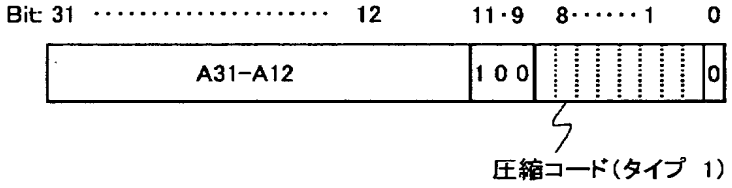
(a)辞書ベースアドレス



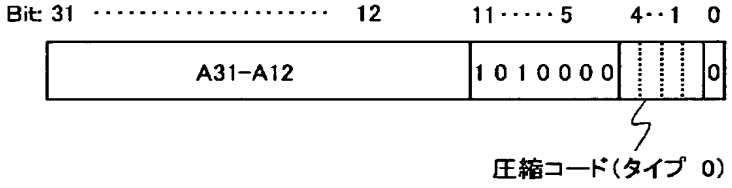
(b)辞書(タイプ 2)アドレス



(c)辞書(タイプ 1)アドレス

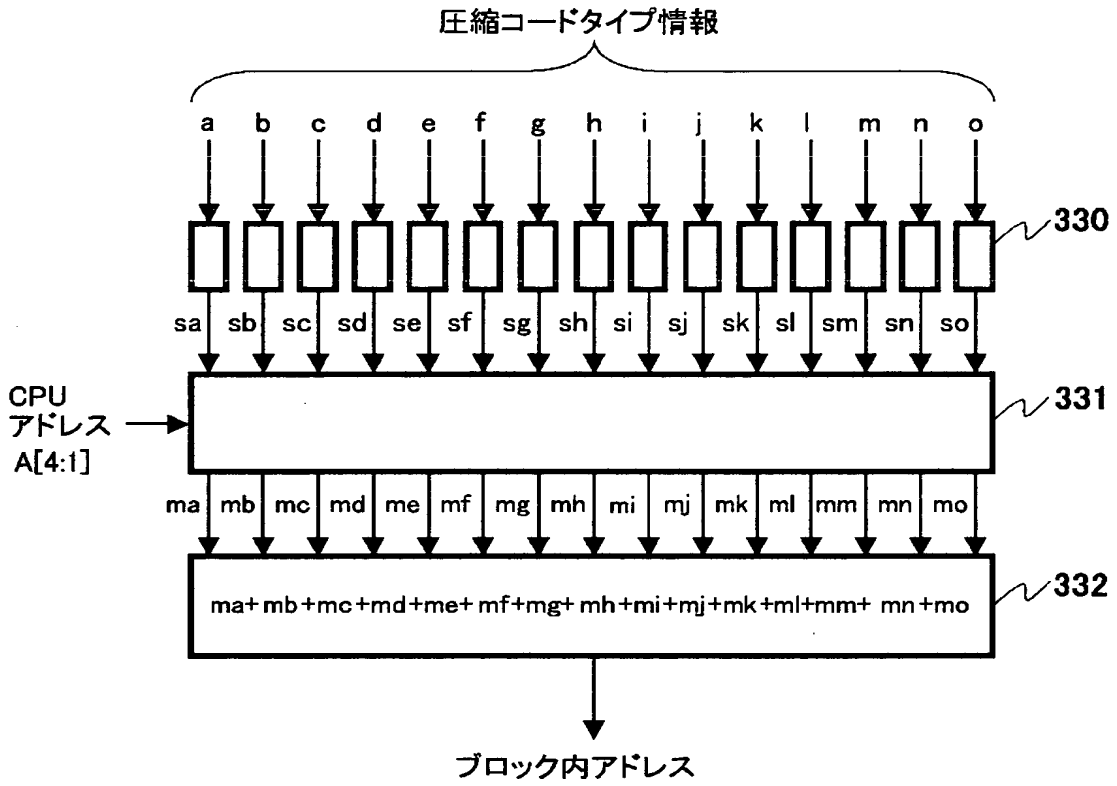


(d)辞書(タイプ 0)アドレス



【図１１】

図11



【図１２】

図12

入力i	出力o
00	0010
01	0100
10	0101
11	1000

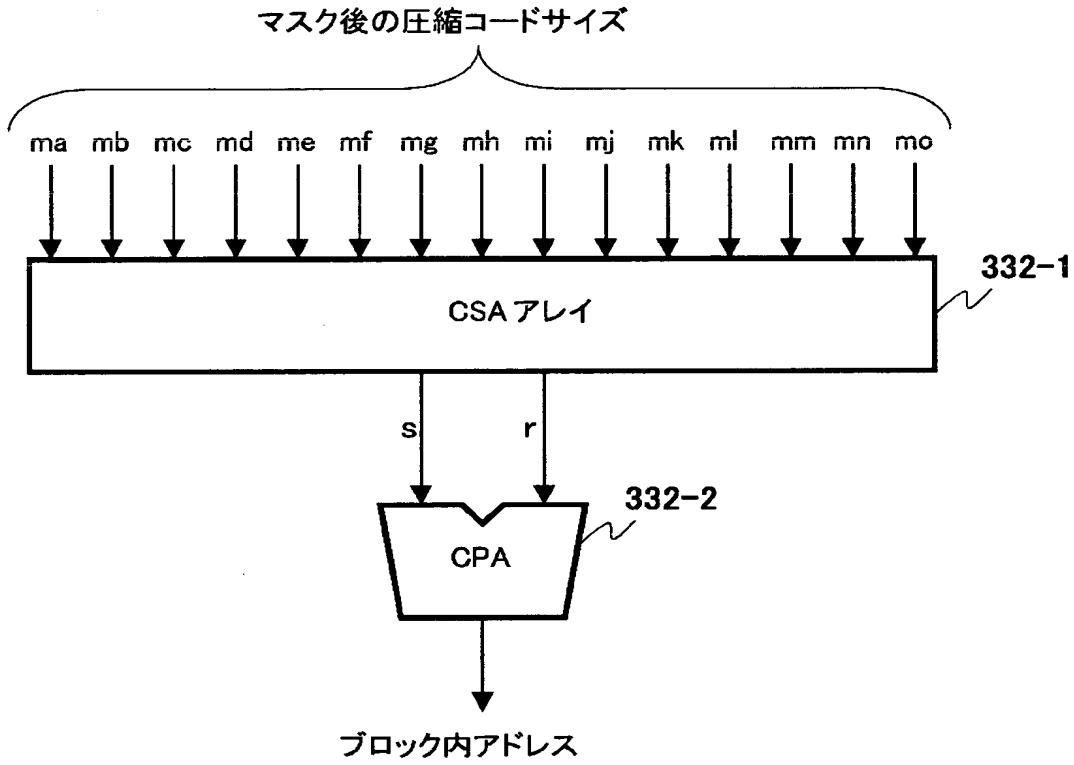
【図１３】

図13

A[4:1]	ma	mb	mc	md	me	mf	mg	mh	mi	mj	mk	ml	mm	mn	mo
0000	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0001	sa	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0010	sa	sb	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0011	sa	sb	sc	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0100	sa	sb	sc	sd	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0101	sa	sb	sc	sd	se	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
0110	sa	sb	sc	sd	se	sf	Z	Z	Z	Z	Z	Z	Z	Z	Z
0111	sa	sb	sc	sd	se	sf	sg	Z	Z	Z	Z	Z	Z	Z	Z
1000	sa	sb	sc	sd	se	sf	sg	sh	Z	Z	Z	Z	Z	Z	Z
1001	sa	sb	sc	sd	se	sf	sg	sh	si	Z	Z	Z	Z	Z	Z
1010	sa	sb	sc	sd	se	sf	sg	sh	si	sj	Z	Z	Z	Z	Z
1011	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	Z	Z	Z	Z
1100	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	Z	Z	Z
1101	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	sm	Z	
1110	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	sm	sn	
1111	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	sm	sn	so

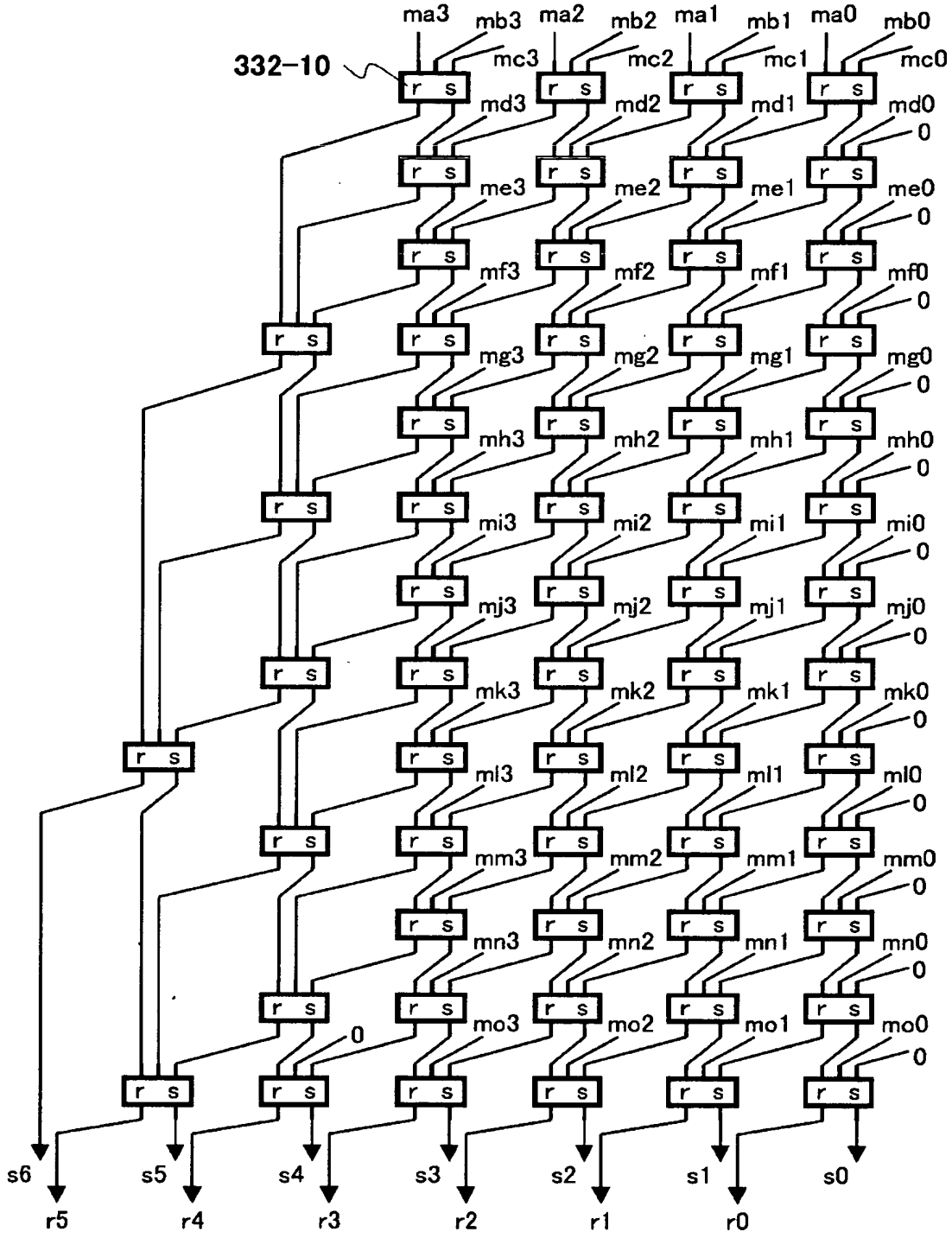
【図１４】

図14


マスク後の圧縮コードサイズ
ma mb mc md me mf mg mh mi mj mk ml mm mn mo
332-1
CSAアレイ
s
r
332-2
CPA
ブロック内アドレス

【図１５】

図15


332-10
ma3 mb3 ma2 mb2 ma1 mb1 ma0 mb0
mc3 mc2 mc1 mc0
md3 md2 md1 md0
me3 me2 me1 me0
mf3 mf2 mf1 mf0
mg3 mg2 mg1 mg0
mh3 mh2 mh1 mh0
mi3 mi2 mi1 mi0
mj3 mj2 mj1 mj0
mk3 mk2 mk1 mk0
ml3 ml2 ml1 ml0
mm3 mm2 mm1 mm0
mn3 mn2 mn1 mn0
mo3 mo2 mo1 mo0
0
r s
s6 s5 s4 s3 s2 s1 s0
r5 r4 r3 r2 r1 r0

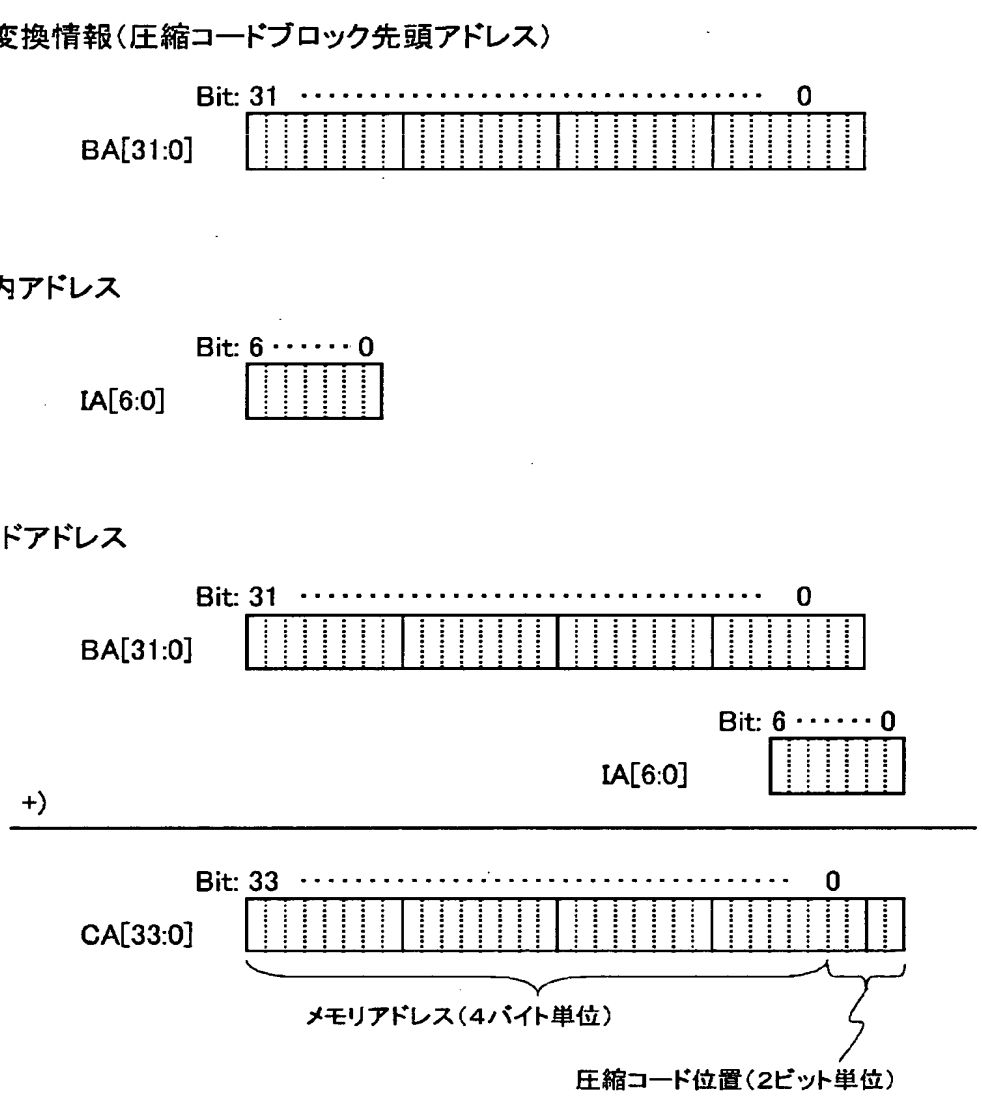
【図１６】

図16

i0	i1	i2	r	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

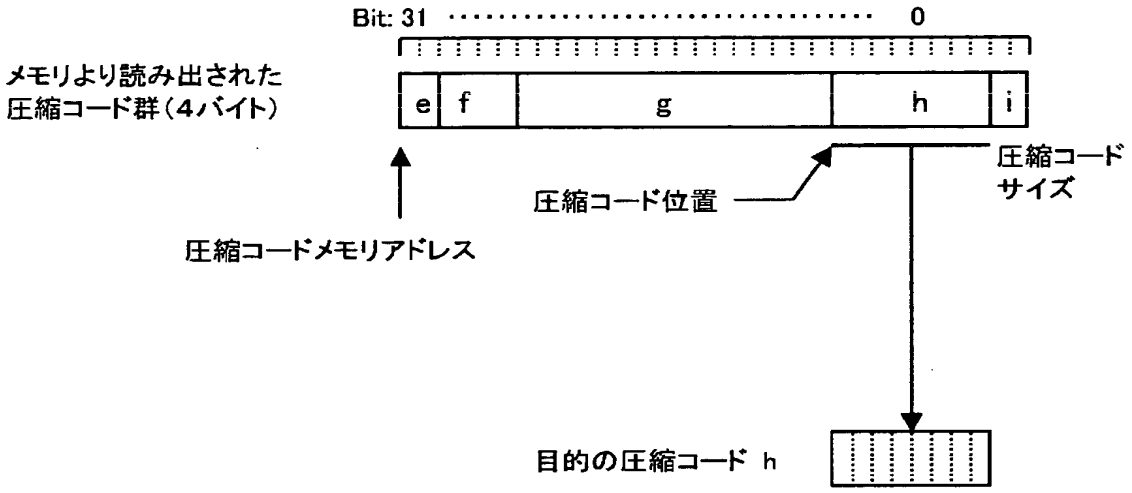
【図１７】

図17


(a)アドレス変換情報(圧縮コードブロック先頭アドレス)
Bit: 31
0
BA[31:0]
(b)ブロック内アドレス
Bit: 6
0
IA[6:0]
(c)圧縮コードアドレス
Bit: 31
0
BA[31:0]
Bit: 6
0
IA[6:0]
+)
Bit: 33
0
CA[33:0]
メモリアドレス(4バイト単位)
圧縮コード位置(2ビット単位)

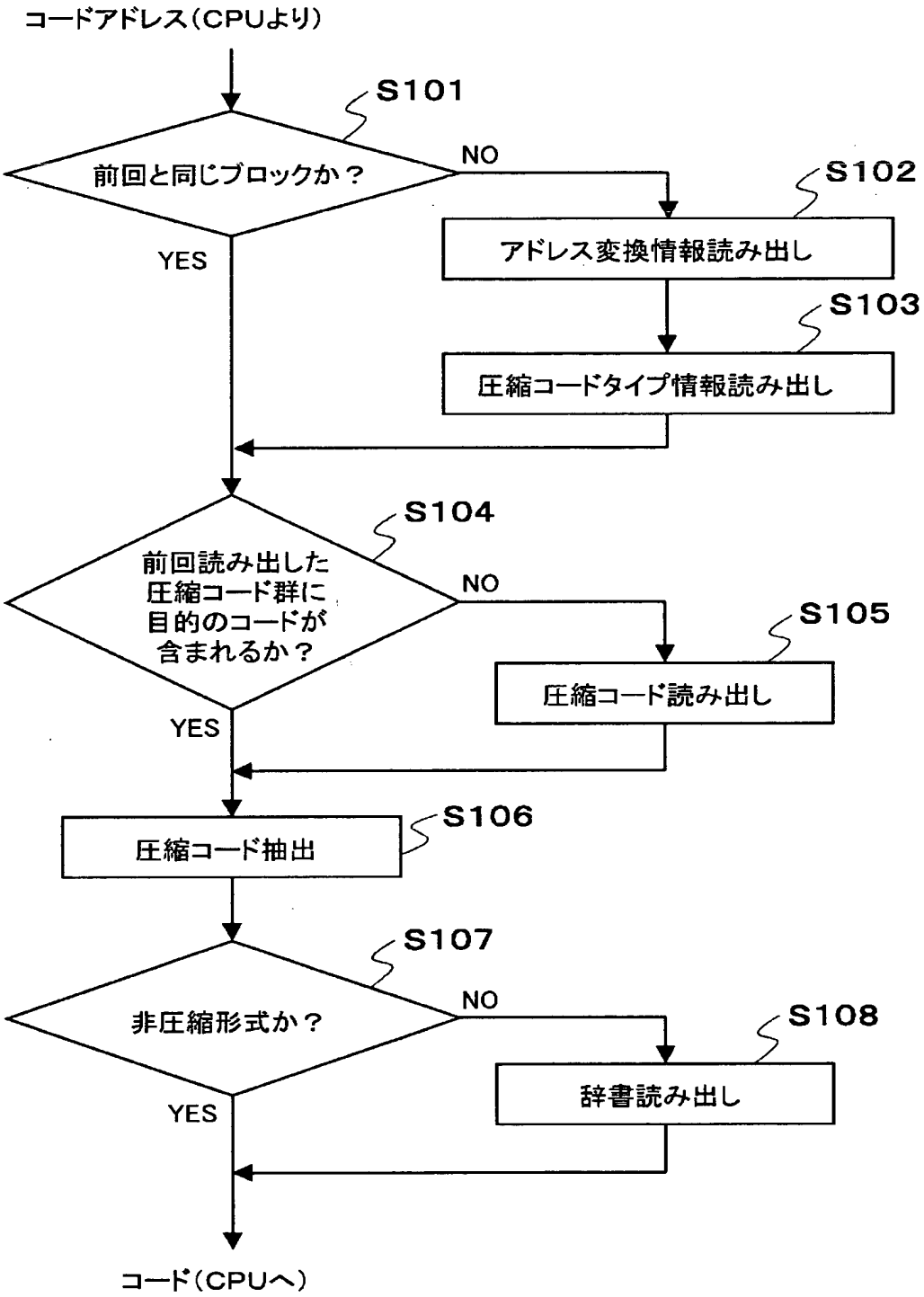
【図１８】

図18


Bit: 31
0
メモリより読み出された
圧縮コード群（4バイト）
e
f
g
h
i
圧縮コード
サイズ
圧縮コード位置
圧縮コードメモリアドレス
目的の圧縮コード h

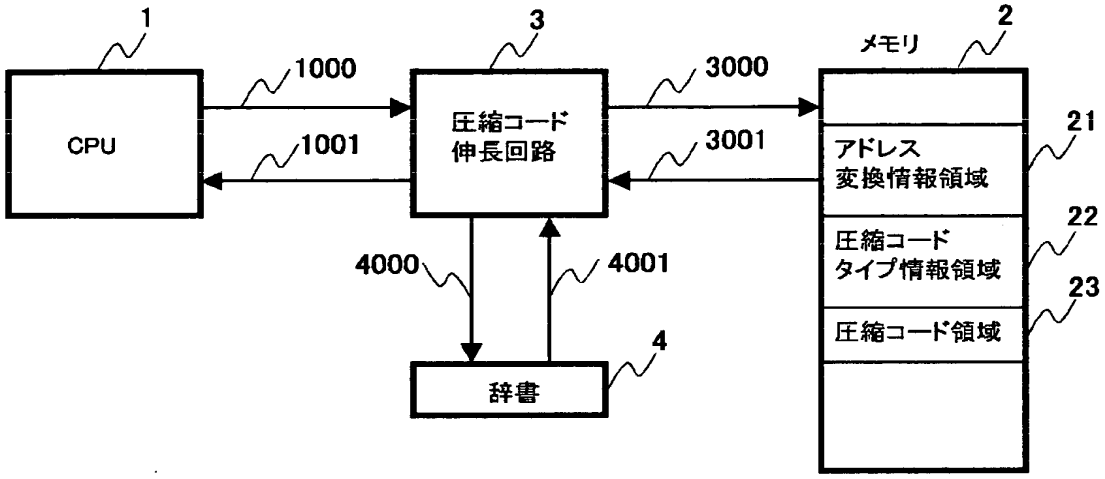
【図１９】

図19


コードアドレス(CPUより)
S101
前回と同じブロックか？
NO
YES
S102
アドレス変換情報読み出し
S103
圧縮コードタイプ情報読み出し
S104
前回読み出した圧縮コード群に目的のコードが含まれるか？
NO
YES
S105
圧縮コード読み出し
S106
圧縮コード抽出
S107
非圧縮形式か？
NO
YES
S108
辞書読み出し
コード(CPUへ)

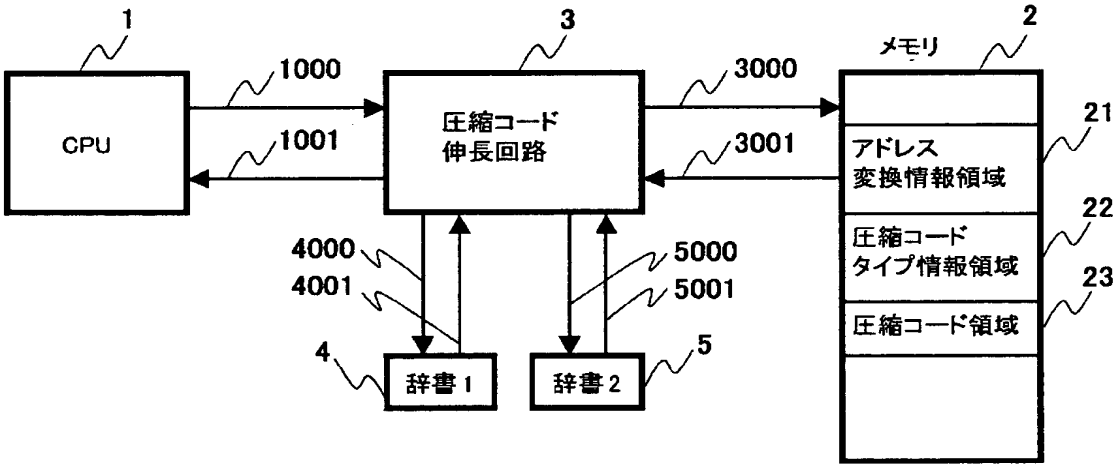
【図２０】

図20



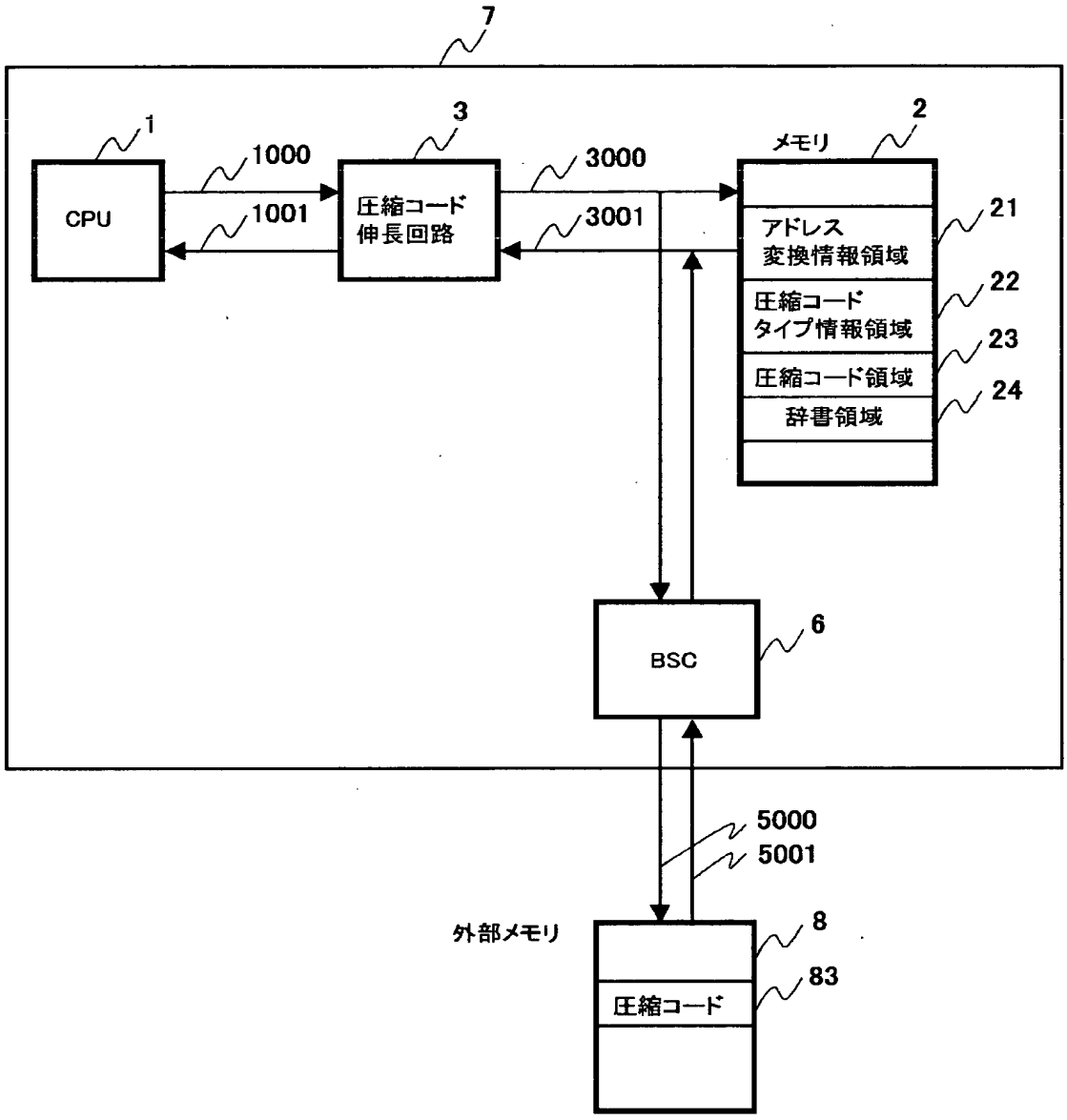
【図２１】

図21



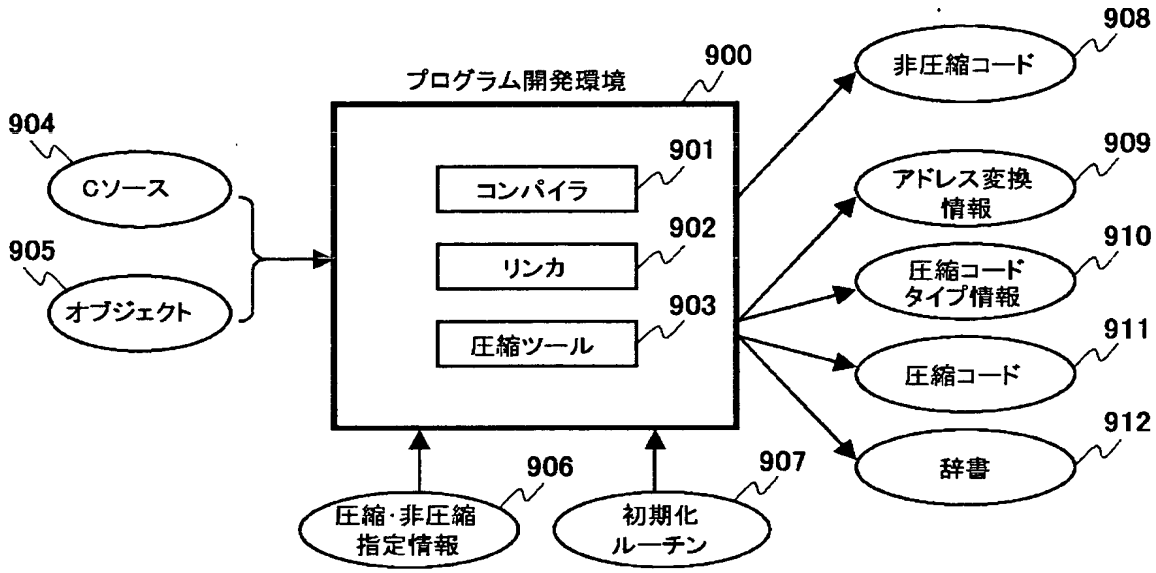
【図２２】

図22


7
1
CPU
1000
1001
3
圧縮コード
伸長回路
3000
3001
2
メモリ
アドレス
変換情報領域
21
圧縮コード
タイプ情報領域
22
圧縮コード領域
23
辞書領域
24
BSC
6
5000
5001
外部メモリ
8
83
圧縮コード

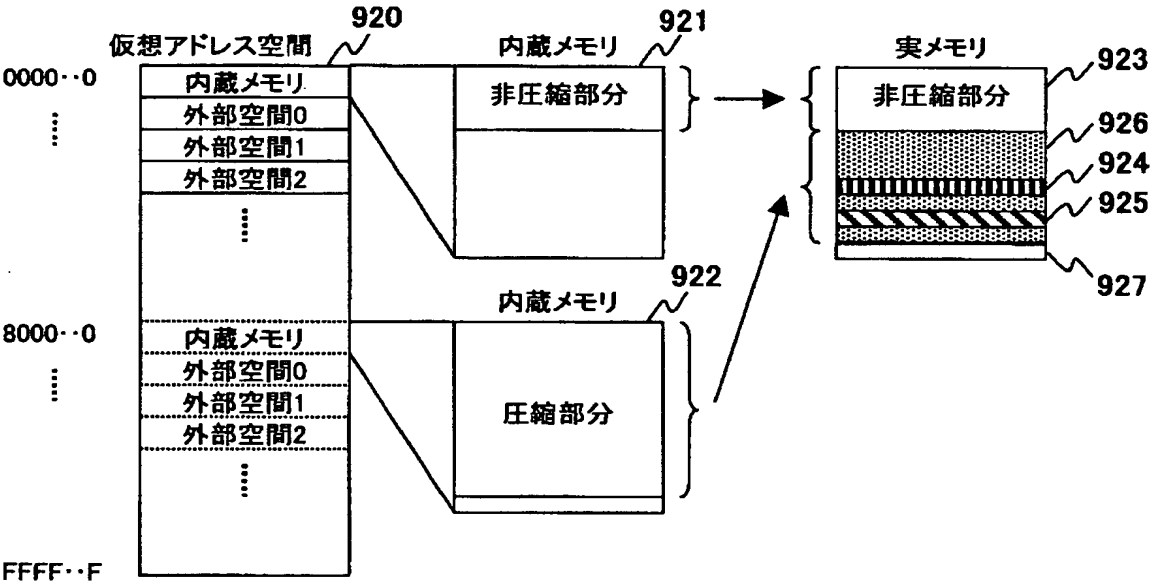
【図２３】

図23



【図２４】

図24



【書類名】　要約書

【要約】

【課題】　マイコン等の組み込み機器に好適なコード圧縮技術を提供する。

【解決手段】　プログラムのコードを可変長コードに変換した圧縮コードと、グループ化されたプログラムのコードについて、各グループの開始アドレスを特定するためのアドレス変換情報と、各グループごとに、グループに含まれる各圧縮コードのコード長を特定するための圧縮コードタイプ情報とをメモリに記憶し、ＣＰＵが出力するコードアドレスから対応する圧縮コードアドレスを直接計算できるように構成することで、マイコン等の組み込み機器に好適なコード圧縮が達成される。

【選択図】　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所